

Remarks

Applicants respectfully request reconsideration of the present application in view of the foregoing amendments and the following remarks.

Examiner Interview Summary

Applicants thank the Examiner for holding a telephone interview with the undersigned representative on January 14, 2009. The interview addressed objections to claim 1 under 35 U.S.C. §§ 101 and 103. Although the participants did not reach a specific agreement, the interview was helpful in advancing prosecution.

Applied Art

The Action applies Tyrrell et al., “CSP Methods for Identifying Atomic Actions in the Design of Fault Tolerant Concurrent Systems” (Tyrrell) and Reps et al., “Precise Interprocedural Dataflow Analysis via Graph Reachability” (Reps).

Claim Amendments

This response:

- amends claim 1 to recite “identified atomically modelable actions” instead of “atomically modelable actions” (see the original specification at, for example, page 2, lines 11-14);
- amends claims 1 and 11 to recite “at a first point in the reachability analysis” and “at a second point in the reachability analysis” (see the original specification at, for example, page 8, line 22 – page 9, line 5);

- amends claim 1 to recite “generating, by the computer” instead of “generating” and amends claim 11 to recite “the computer generating” instead of “generating” (see the original specification at, for example, page 42, lines 7-9);
- amends claim 1 to recite “storing the at least one partial procedure summary of the procedure of multithreaded software on a second computer readable medium,” amends claim 11 to recite “storing in a second computer-readable medium at least some of the plurality of generated partial procedure summaries for the multithreaded software,” and amends claim 15 to recite “a second computer-readable medium containing instructions configured to cause the microprocessor to store in a third computer-readable medium an indication of one or more programming flaws identified by the model checker” (see the original specification at, for example, page 13, lines 3-6);
- amends claim 1 to recite “the resulting state of the at least one partial procedure summary is used as a state of the procedure” and amends claim 15 to recite “a reachability analyzer operable to employ, upon reaching the sets of actions at least a second time during a reachability analysis, one or more of the partial procedure summaries” (see the original specification at, for example, page 9, lines 1-5);
- amends claim 15 to recite a “system” instead of a “computer-implemented system” and to recite “a microprocessor” and “a first computer-readable medium containing instructions configured to reconfigure the microprocessor to act as at least a portion of a model checker” (see the original specification at, for example, page 42, lines 7-20); and
- adds new independent claim 25.

Claim Rejections under 35 U.S.C. § 112

Applicants thank the Examiner for indicating that the rejection of claim 23 under § 112, ¶ 2, has been withdrawn. See Action at page 2. The Action, in discussing claim 23 at page 13, still refers to a “112 2nd rejection.” Applicants respectfully presume this is an error. In any case, Applicants note that claim 23 satisfies § 112, ¶ 2, for at least the reasons set forth at pages 9-10 of the response filed August 25, 2008. Applicants respectfully request the withdrawal of any remaining rejections under § 112, ¶ 2.

Claim Rejections under 35 U.S.C. § 101

The Action rejects claims 1, 3-16, 18, 21, 23 and 24 under 35 U.S.C. § 101. Applicants respectfully traverse all rejections under § 101. The rejection of claim 24 is moot in view of the cancellation of this claim without prejudice.

The Action at page 4 refers to MPEP 2106 and states:

The claimed invention as a whole must accomplish a practical application. That is, it must produce a “useful, concrete and tangible result” State Street 149 F.3d at 1373, 47 USPQ2d at 1601-02. A process that consists solely of the manipulation of an abstract idea is not concrete or tangibles.

Applicants respectfully note that Federal Circuit recently held:

[W]hile looking for “a useful, concrete and tangible result” may in many instances provide useful indications of whether a claim is drawn to a fundamental principle or a practical application of such a principle, *that inquiry is insufficient to determine whether a claim is patent-eligible under § 101*. . . . Therefore, we also conclude that the “useful, concrete and tangible result” inquiry is inadequate. . . .

In re Bilski, ___ F.3d ___ (Fed. Cir. 2008) (Slip Op. at 20) (emphasis added).¹ Instead, the *Bilski* court explained:

¹ The *Bilski* court added: “As a result, those portions of our opinions in *State Street* and *AT&T* relying solely on a ‘useful, concrete and tangible result’ analysis should no longer be relied on.” See Slip Op. at fn. 19.

The Supreme Court, however, has enunciated a *definitive test* to determine whether a process claim is tailored narrowly enough to encompass only a particular application of a fundamental principle rather than to pre-empt the principle itself. *A claimed process is surely patent-eligible under § 101 if: (1) it is tied to a particular machine or apparatus, or (2) it transforms a particular article into a different state or thing.*

Id., Slip Op. at 10 (internal citations omitted; emphases added). The *Bilski* court refers to this “definitive test” as the “machine-or-transformation test.” *Id.*, Slip Op. at 12.

The method acts recited in claim 1 satisfy *both* prongs of the machine-or-transformation test. For example, the method acts cause “a computer to perform a method of generating a partial procedure summary of a procedure of multithreaded software.” One such method act recites “generating, *by the computer*, the at least one partial procedure summary of the procedure.” Thus, the method acts are associated with a particular machine, namely, a programmed computer.

The Action states at page 5 that the claims do not “require a ‘particular machine’ since any general purpose computer can carry out the analysis.” Applicants respectfully disagree with this assertion and note that the Patent and Trademark Office is bound by the Federal Circuit’s interpretation of *Benson*:

The reconsideration Board majority also erred in its reasoning that [apparatus] claim 15 is unpatentable merely because it “reads on a general purpose digital computer ‘means’ to perform the various steps under program control.” ... Alappat admits that claim 15 would read on a *general purpose computer programmed to carry out the claimed invention*, but argues that this alone also does not justify holding claim 15 unpatentable as directed to nonstatutory subject matter. We agree. *We have held that such programming creates a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software.*

Under the Board majority’s reasoning, a programmed general purpose computer could never be viewed as patentable subject matter under Section 101. This reasoning is without basis in the law. The Supreme Court has never held that a programmed computer may never be entitled to patent protection. Indeed, the Benson court specifically stated that its decision therein did not preclude “a patent for any program servicing a computer.” Consequently, a computer operating pursuant to software may

represent patentable subject matter, provided, of course, that the claimed subject matter meets all of the other requirements of Title 35.

In re Alappat, 33 F.3d 1526, 1544, 31 USPQ2d 1545, 1558 (Fed. Cir. 1994) (internal citations omitted; emphases added).

The method acts recited in claim 1 also satisfy the second prong of the machine-or-transformation test by transforming a particular article to a different state or thing. For example, the method acts recited in claim 1 store “the at least one partial procedure summary of the procedure of multithreaded software on a second computer readable medium.” This transforms the second computer readable medium to a different state or thing.

For at least these reasons, the method acts recited in claim 1 satisfy *both* prongs of the machine-or-transformation test.

The Action at page 4 states that “the claimed invention pre-empts the field in which it takes part.” The Action further states:

Here the claims are so abstract and sweeping as to cover both known and unknown uses of multithreaded software procedure analysis. The end use may (i) encompass any type of multithreaded program (See specification page 3, lines 11-14) and (2) can be performed through any existing machinery or future-devised machinery or without any apparatus. . . . The claims listed above lack any particular application, and could potentially be applied to any field that encompasses data collection and modeling and could possibly be applied to in the future and the past, which pre-empts the industry from further research in the area.

See page 5. Applicants respectfully disagree. First, as quoted above, the *Bilski* court explained that the machine-or-transformation test is the Supreme Court’s “*definitive test to determine whether a process claim is tailored narrowly enough to encompass only a particular application of a fundamental principle rather than to pre-empt the principle itself.*” *In re Bilski*, Slip Op. at 10 (emphases added). Since the method acts recited in claim 1 satisfy this test, claim 1 cannot pre-empt any fundamental principle. Additionally, the claim language shows that claim 1

encompasses only a *particular application* of a fundamental principle. For example, claim 1 recites:

A computer program product embodied on a first computer-readable medium and comprising code that, when executed, causes a computer to perform a method of generating a partial procedure summary of a procedure of multithreaded software, wherein the procedure performs a plurality of actions when executed, the method comprising:

 during a reachability analysis of at least a portion of the multithreaded software, reaching the procedure . . .

 when the procedure is reached at the first point in the reachability analysis,

 1) identifying at least one plurality of the actions within the procedure . . .

 2) generating, by the computer, the at least one partial procedure summary of the procedure from the identified atomically modelable actions . . .

 storing the at least one partial procedure summary of the procedure of multithreaded software on a second computer readable medium. . . .

As claim 1 recites a “*computer program product embodied on a first computer-readable medium and comprising code that, when executed, causes a computer to perform a method,*” “*generating, by the computer*” and “*storing the at least one partial procedure summary of the procedure of multithreaded software on a second computer readable medium,*” it is incorrect to say that claim 1 “pre-empts the field in which it takes part,” “can be performed . . . without any apparatus,” or “lack[s] any particular application.” For at least these reasons, claim 1 does not, as the Action suggests at page 5, “patent every ‘substantial practical application’ of an idea, law of nature or natural phenomena.”

The Action at page 4 also states, “A claim that recites a computer that solely calculates a mathematical formula is not statutory.” However, as explained above, the method acts recited in claim 1 do not “solely calculate[] a mathematical formula.” For example, the method acts recite “*storing the at least one partial procedure summary of the procedure of multithreaded software on a second computer readable medium.*”

Finally, the Board of Patent Appeals and Interferences recently (after the Federal Circuit’s *Bilski* ruling) held that a claim reciting a “computer program product, comprising a computer usable medium having a computer readable program code embodied therein” satisfies 35 U.S.C. § 101.² See *Ex parte Bo Li*, Appeal 2008-1213, Application No. 10/463,287.

For at least these reasons, claim 1 satisfies § 101.

Claim 11 recites, “A *computer program product embodied on a first computer-readable medium* and comprising code that, when executed, *causes a computer to perform a method* of modeling multithreaded software, the method comprising . . . based on the analyzing, *the computer generating a plurality of partial procedure summaries* for the multithreaded software . . . *storing in a second computer-readable medium at least some of the plurality of generated partial procedure summaries* for the multithreaded software. . . .” For at least these reasons, claim 11 satisfies both prongs of the machine-or-transformation test, and claim 11 satisfies § 101.

Claim 15 recites, “A system . . . comprising: a *microprocessor; a first computer-readable medium* containing instructions configured to reconfigure the microprocessor to act as at least a

² The relevant claim in *Bo Li* reads:

42. A computer program product, comprising a computer usable medium having a computer readable program code embodied therein, said computer readable program code adapted to be executed to implement a method for generating a report, said method comprising:

providing a system, wherein the system comprises distinct software modules, and wherein the distinct software modules comprise a logic processing module, a configuration file processing module, a data organization module, and a data display organization module;

parsing a configuration file into definition data that specifies: a data organization of the report, a display organization of the report, and at least one data source comprising report data to be used for generating the report, and wherein said parsing is performed by the configuration file processing module in response to being called by the logic processing module;

extracting the report data from the at least one data source, wherein said extracting is performed by the data organization module in response to being called by the logic processing module;

receiving, by the logic processing module, the definition data from the configuration file processing module and the extracted report data from the data organization module; and

organizing, by the data display organization module in response to being called by the logic processing module, a data display organization of the report, wherein said organizing comprises utilizing the definition data received by the logic processing module and the extracted report data received by the logic processing module.

portion of a model checker . . . and *a second computer-readable medium* containing instructions configured to cause the microprocessor to store in *a third computer-readable medium* an indication of one or more programming flaws identified by the model checker.” For at least these reasons, claim 15 satisfies both prongs of the machine-or-transformation test, and claim 15 satisfies § 101.

Claims 3-10, 21 and 23 depend from claim 1; claims 12-14 depend from claim 11; and claims 16 and 18 depend from claim 15. All of these dependent claims satisfy § 101 for at least reasons similar to those stated above for their respective parent claims.

Applicants respectfully request withdrawal of all rejections under 35 U.S.C. § 101.

Claim Rejections under 35 U.S.C. § 103

The Action rejects claims 1, 3-16, 18, 21, 23 and 24 under 35 U.S.C. § 103(a) over Tyrrell in view of Reps. Applicants respectfully traverse these rejections. The rejection of claim 24 is moot in view of the cancellation of this claim without prejudice.

Independent Claim 1

Claim 1 recites:

A computer program product embodied on a first computer-readable medium and comprising code that, when executed, causes a computer to perform a method of generating a partial procedure summary of a procedure of multithreaded software, wherein the procedure performs a plurality of actions when executed, the method comprising:

- during a reachability analysis of at least a portion of the multithreaded software, reaching the procedure at a first point in the reachability analysis;
- when the procedure is reached at the first point in the reachability analysis,
 - 1) identifying at least one plurality of the actions within the procedure as atomically modelable with respect to multithreaded execution of the procedure as

atomically modelable actions, wherein the atomically modelable actions are not subject to interruption by other threads, and

2) generating, by the computer, the at least one partial procedure summary of the procedure from the identified atomically modelable actions . . .

storing the at least one partial procedure summary . . .

during continuation of the reachability analysis, at a second point in the reachability analysis, reaching the procedure a second time; and

wherein if an initial state of the procedure reached the second time is the same as the initial state of the at least one partial procedure summary, then the resulting state of the at least one partial procedure summary is used as a state of the procedure in the reachability analysis.

Tyrrell and Reps do not, singularly or in combination, teach or suggest the computer program product of claim 1. For example, these references do not teach or suggest “if an initial state of the procedure reached the second time is the same as the initial state of the at least one partial procedure summary, then the resulting state of the at least one partial procedure summary is used as a state of the procedure in the reachability analysis.”

Instead, as the Action notes at page 7, Tyrrell discloses:

Any attempt to incorporate an entry line and an exit line at arbitrary locations in a concurrent system is unlikely to lead to a properly formed recovery mechanism. It is necessary to identify a boundary within the state space of the complete set of processes across which error propagation by communication is prevented. Clearly, this boundary will be the boundary of an atomic action, since such a boundary of necessity prohibits the passing of information to any process not involved in the atomic action and similarly embraces all interacting processes within the atomic action. Recovery mechanisms can be nested systematically in the same hierarchical fashion as atomic actions. If this duality is not imposed, then should the system attempt to backtrack and recover in response to a fault, progressive collapse by the domino effect can occur.

See Tyrrell at page 630, col. 1, ¶ 4 (emphasis in original; internal citation omitted). However, Tyrrell’s description of “a boundary within the state space of the complete set of processes” and “recovery mechanisms” does not teach or suggest, as recited in claim 1, *reaching a state a second time* and that “the resulting state of the at least one partial procedure summary is *used as a state of the procedure in the reachability analysis.*” Also, Tyrrell discloses at page 633, col. 1,

bottom half, “special events” which are “added before the start of each process.” Still, this does not teach or suggest the above features of claim 1.

Reps does not remedy the shortcomings of Tyrrell for claim 1. For example, Reps also fails to teach or suggest “if an initial state of the procedure reached the second time is the same as the initial state of the at least one partial procedure summary, then the resulting state of the at least one partial procedure summary is used as a state of the procedure in the reachability analysis.”

For at least these reasons, claim 1 is allowable over Tyrrell and Reps.

Independent Claim 11

Claim 11 recites:

A computer program product embodied on a first computer-readable medium and comprising code that, when executed, causes a computer to perform a method of modeling multithreaded software, the method comprising:
performing a reachability analysis of the multithreaded software;
during the reachability analysis, reaching a procedure . . .
analyzing actions of the multithreaded software within the procedure . . .
based on the analyzing, the computer generating a plurality of partial procedure summaries for the multithreaded software . . . and
during the reachability analysis, reaching the procedure at a second point in the reachability analysis and reusing the plurality of partial procedure summaries to determine actions executable atomically. . . .

Tyrrell and Reps do not, singularly or in combination, teach or suggest the computer program product of claim 11. For example, these references do not teach or suggest “reaching the procedure at a second point in the reachability analysis and reusing the plurality of partial procedure summaries to determine actions executable atomically.” Instead, Tyrrell discloses at page 633, col. 1, bottom half, “special events” which are “added before the start of each process.” Reps discloses “summary edges” and “iterations” at page 54, section 4, but these do

not teach or suggest “reaching the procedure at a second point in the reachability analysis and reusing the plurality of partial procedure summaries to determine actions executable atomically.” For at least these reasons, claim 11 is allowable over Reps and Tyrrell.

Independent Claim 15

Claim 15 recites:

A system for modeling multithreaded software, the system comprising:
a microprocessor;
a first computer-readable medium containing instructions configured to reconfigure the microprocessor to act as at least a portion of a model checker . . . comprising:
the model of the multithreaded software, wherein the model comprises a plurality of partial procedure summaries modeling beginning states and ending states of partial summaries of procedures within the multithreaded software during multithreaded execution of the multithreaded software, the partial procedure summaries comprising the start and end states of sets of . . . and
a reachability analyzer operable to employ, upon reaching the sets of actions at least a second time during a reachability analysis, one or more of the partial procedure summaries instead of the sets of actions to generate modeled states of the software. . . .

Tyrrell and Reps do not teach or suggest such a system. For example, as Applicants similarly explained above for claims 1 and 11, Tyrrell and Reps do not teach or suggest “a reachability analyzer operable to employ, upon reaching the sets of actions at least a second time during a reachability analysis, one or more of the partial procedure summaries instead of the sets of actions to generate modeled states of the software.” For at least these reasons, claim 15 is allowable.

Dependent Claims

Claims 3-10, 21 and 23 depend from claim 1; claims 12-14 depend from claim 11; and claims 16 and 18 depend from claim 15. Each of these claims is allowable over Tyrrell and Reps

for at least the reasons stated above for their respective parent claims, as well as for the unique combinations of features recited therein.

Applicants respectfully request withdrawal of all rejections under 35 U.S.C. § 103(a).

New Claim

New claim 25 is independent. Claim 25 is allowable for at least the reason that the prior art does not teach or suggest “identifying a sequence of the actions within the procedure having one or more right movers followed by an atomic action followed by one or more left movers, the identifying further comprising identifying a transaction boundary within the plurality of actions using a phase variable, the phase variable indicating whether a thread is modeled as executing at least one of the one or more left movers of the identified sequence of actions or at least one of the one or more right movers of the identified sequence of actions.”

Conclusion

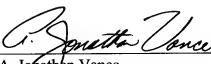
In view of the foregoing amendments and remarks, Applicants respectfully submit that the Application is in condition for allowance. Should any issues remain, Applicants ask the Examiner to contact the undersigned representative.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By


A. Jonathan Vance
Registration No. 56,258